WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Wednesday, May 05, 2004

Hide?	Set Name	Query	Hit Count
		DB=USPT,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR	
☐	L28	(boot near3 perform$4 near3 flag$4)	7
☐	L27	L16 and l24	11
☐	L26	l8.ab. and L24	3
☐	L25	l8 and L24	56
☐	L24	l18 or l19 or l20 or l21 or l22 or L23	4106
☐	L23	709/222.ccls.	328
☐	L22	709/221.ccls.	493
☐	L21	709/220.ccls.	727
☐	L20	709/219.ccls.	1472
☐	L19	713/2.ccls.	815
☐	L18	713/1.ccls.	970
☐	L17	L16.ab.	3
☐	L16	(warm adj reboot$4)	42
☐	L15	l8.ab.	19
☐	L14	l11.ab.	4
☐	L13	reboot$4 near3 (file adj server)	4
☐	L12	reboot$4 with (file adj server)	7
☐	L11	reboot$4 same (file adj server)	28
☐	L10	L8 same (file adj server)	0
☐	L9	L8 same performance	4
☐	L8	reboot$4 same flag$4	230
☐	L7	(boot$4 near5 flag$4) and (indicat$4 near3 performance)	1
☐	L6	(boot$4 near5 flag$4) same (indicat$4 near3 performance)	0
☐	L5	(warm adj boot$4) and (skip$4 near3 operation)	1
☐	L4	(warm adj boot$4) same (skip$4 near3 operation)	0
☐	L3	(full adj reboot$4) and (warm adj reboot$4)	1
☐	L2	(full adj reboot$4) same (warm adj reboot$4)	0
☐	L1	(fast near2 reboot$4)	9

END OF SEARCH HISTORY

L9: Entry 1 of 4 File: USPT Aug 22, 2000

DOCUMENT-IDENTIFIER: US 6108100 A
TITLE: Apparatus and method for end-user performance upgrade

Detailed Description Text (33):
FIG. 6 illustrates a method of pre-flagging expansion memory units 34. It is possible for a manufacturer or distributor of the electronic devices to pre-flag memory expansion units 34 as being the default run location. For example, a customer may complain that system performance is too slow. In this case, the manufacturer could ship a fast expansion memory unit 34 that is pre-flagged as the default run location. When the customer receives the expansion memory unit 34, they can simply install the new expansion memory unit 34 and reboot to immediately realize system performance improvements.

L9: Entry 2 of 4 File: USPT Dec 31, 1996

DOCUMENT-IDENTIFIER: US 5590376 A
TITLE: Apparatus and method for detecting and updating status of removable storage system using shared interrupt and software flag

Detailed Description Text (3):
The present invention provides flexibility and fast performance by employing a scheme for setting a flag in the memory of a computer system when a removable storage medium has been removed from the computer system. The removable storage medium thus need not be accessed prior to commencing each media operation to determine whether the removable storage medium has been changed since the last media operation. The present invention also allows removable media of various sizes and formats to be interchanged "on the fly" without the need to reboot or manually reconfigure the computer system. In particular, the present invention contemplates a scheme for allowing existing removable media storage mechanisms to communicate with the present invention, thereby allowing these existing removable storage medium mechanisms to accept media of varying formats and storage capacities and to take advantage of the fast response time offered by the present invention.

L9: Entry 3 of 4 File: USPT Mar 14, 1995

DOCUMENT-IDENTIFIER: US 5398277 A
TITLE: Flexible multiprocessor alarm data processing system

Detailed Description Text (77):
As part of its supervisory function over system 120, system supervisor 130 will periodically examine the performance of processors 122 to be sure no processor is failing or defective. Thus, system supervisor 130 will periodically, such as every 3-30 seconds, and also every time the system supervisor completes any task it was performing, examine processors status records 60 for two items. First, the Function ID in the status record as written by the processor 122 is compared to the working function program previously named by system supervisor 130 in the AUTOEXEC.BAT file for that processor. Second, the time stamp is examined to see if it is less than 2 minutes old. If either test is not answered in the affirmative, then the processor 122 may be faulty. In that event, the reboot flag for that processor 122 is set, but the working function program in the AUTOEXEC.BAT file is given the same name it was previously supposed to have so that upon rebooting, the processor will run the same function it was previously supposed to run.

First Hit

End of Result Set

Generate Collection | Print

L9: Entry 4 of 4 File: TDBD Apr 1, 1994

DOCUMENT-IDENTIFIER: NA9404153
TITLE: Unconditional Installation of SCSI Device Drivers

Disclosure Text (1):
Disclosed is the unconditional installation of a device driver using the information stored in BIOS to determine the kind and number of SCSI devices present in the system. If an option flag used by this installation feature is set, the device driver installs itself, and installs all registered devices, even if the devices do not respond to commands. - Without this feature, device drivers do not install if no associated devices are found during the initialization of the device driver. For external devices, the most common reason a device is not found is that it has not been powered on at this time. When this occurs, the user must either turn the device on and reboot the system to repeat the initialization process, or the user must continue to operate the system without using the device. This alternative may cause additional problems to occur if drive letter assignments have changed due to the missing device. Batch files depending on drive letter knowledge may fail. These problems are exacerbated if this condition occurs in a server. - Machine configuration information is maintained in NVRAM for use at power-on by the POST (Power-On Self-Test) process. When the system is started, the POST test determines the current configuration, which is compared to the expected configuration for which information is stored. If a discrepancy is found, the user is advised to run the Setup Utility to install new devices, or to test possibly defective devices. - However, the unconditional installation feature of the Setup Utility provides an option to enable or to disable presence checking of individual SCSI devices. With presence checking enabled, if a device listed in the stored configuration data does not respond to the POST test, an error is reported. With presence checking disabled, such an error is not reported, so the device is registered with BIOS, to be available for allocation and access through the BIOS interfaces, even though the device is actually not present. When presence checking is disabled by the user, POST and BIOS lay necessary groundwork for the unconditional installation of the device by the device driver, which determines system configuration from the information stored in BIOS. At this point, the device driver can install devices not responding to commands, without verifying the operation of the device before installation. The user indicates, by an option flag, that the device driver is to install all devices registered in BIOS, and to install itself, even if these devices do not respond to the usual SCSI device inquiry or to other commands. - The device driver initialization code sets the device-installed flag to true or false when a device driver is initially loaded into the system. If the device is found to be present at initialization time, the device- installed flag is set to true, resulting in the traditional installation of the device driver. However, if the unconditional-install flag has been set by the user, and if the device is not present during initialization, the driver proceeds as if the device were present, installing the device in its tables, and reporting to the operating system that the device is present, so that a drive letter will be reserved for it. However, under these condi- tions, the device driver marks the device-installed flag as false for this device, so that the device driver runtime code will determine whether the device has become active. - At runtime, the behavior of the device is reflected by the device driver according to the changing

status of the device. The device driver determines the operability and level of support for the device quickly and transparently. If the device remains unresponsive, the device driver reports a general-failure error when the system or the user attempts to access the device. However, if the device becomes operable before such an attempt is made, for example, by being powered on by the user, the device can be used. This process requires the device-installed flag to indicate that the device has been installed, a way to determine the installation status at runtime, and installation code that can be executed during runtime. - Checking the device-installed flag every time a command is issued for a device is very inefficient, since the device in question is installed in the vast majority of instances. Fortunately, the Build BPB command occurs whenever such a test of the device-installed flag is needed, being called each time the media is mounted by the file system. The media is mounted when a new disk is inserted into a drive with removable media, or when a device with removable or fixed media is accessed for the first time following the bootup procedure. Since the procedure of mounting a file is both time consuming and infrequent, the test of the device-installed flag, which is performed with the Build BPB command, has a negligible effect on system performance. - During the Build BPB command, if the device driver determines that a functional device has not been installed, the initialization procedures that have been deferred until runtime are performed. The device driver first checks to see if the device has become functional. If it has not become functional, the status of this command is returned to the operating system. If the device has become functional, a device inquiry command is issued to the device. From the inquiry information, the level of support for the device is determined. The vendor identification is saved for use by the format and other commands. A Read Device Capacity command is issued to clear the power-on check condition at the device, the device-installed flag is set to true, and normal Build BPB procedures are continued.

L13: Entry 2 of 4 File: USPT Sep 12, 2000

DOCUMENT-IDENTIFIER: US 6119244 A
TITLE: Coordinating persistent status information with multiple file servers

Detailed Description Text (34):
In a REBOOTING state 240, this file server 110 has control of none of the mass storage devices 120 and is recovering from a service interruption.

Detailed Description Text (99):
In this state, this file server 110 performs no operations, until this file server 110 determines that it reboot.

Detailed Description Text (103):
In the REBOOTING state 240, this file server 110 has control of none of the mass storage devices 120 and is recovering from a service interruption.

Detailed Description Text (105):
If this file server 110 is unable to recover from the service interruption, the REBOOT-FAILED transition 241 is taken and this file server 110 remains in the REBOOTING state 240.

Detailed Description Text (106):
If this file server 110 is able to recover from the service interruption, but the other file server 110 is in the TAKEOVER state 220, the REBOOT-FAILED transition 241 is taken and this file server 110 remains in the REBOOTING state 240. In this case, the other file server 110 controls the mass storage devices 120 normally assigned to this file server 110, and this file server 110 waits for the GIVEBACK-OPERATION transition 221 before re-attempting to recover from the service interruption.

Detailed Description Text (119):
At a step 258, if this file server 110 was in the NORMAL state 210 before entering the REBOOTING state 240 (that is, this file server 110 performed the step 254 and seized only its own mass storage devices 120), it enters the NORMAL state 210.

Detailed Description Text (120):
At a step 258, if this file server 110 was in the TAKEOVER state 220 before entering the REBOOTING state 240 (that is, this file server 110 performed the step 255 and seized all the mass storage devices 120, it enters the TAKEOVER state 220.

L26: Entry 1 of 3 File: USPT Feb 13, 2001

DOCUMENT-IDENTIFIER: US 6189114 B1
TITLE: Data processing system diagnostics

Abstract Text (1):
A system and method provides remote diagnostics testing of a data processing system. Diagnostics testing code is stored in a non-volatile memory in the system. A diagnostic test indicator (e.g., in the form of flag in a CMOS RAM) is settable by a signal from a controlling computer system remote from the data processing system. The signal requests that diagnostics testing is to be performed on the data processing system. When the data processing system is rebooted, the CMOS flag is checked and if found to be set, the diagnostics code is invoked and diagnostic testing is performed. When the diagnostics testing is complete and results have been logged in the non-volatile storage, the code causes the flag to be reset and the data processing system to be rebooted. The results are transferred, on request, to the remote controlling computer system for analysis.

Current US Cross Reference Classification (1):
713/2

L26: Entry 2 of 3 File: USPT Sep 24, 1996

DOCUMENT-IDENTIFIER: US 5559957 A
TITLE: File system for a data storage device having a power fail recovery mechanism for write/replace operations

Abstract Text (1):
The present invention provides a method and apparatus in a data storage device of a data storage system under the control of a microprocessor for preventing a microprocessor stall upon the occurrence of a power failure during read/write operations. During normal operations, files are written to a first storage area of the data storage device where a first flag associated with each file is set when the writing of the respective file has successfully completed. Upon the occurrence of a power failure, a data storage device initialization routine is commenced upon reboot of the microprocessor. During initialization, an analysis phase is begun to generate sequences of events for at least those files not having the first flag set. The events comprise memory operations and associated data and are each re-executable upon interruption of its execution and before execution of another event without modifying results of a previous execution of the respective event. The events generated are then written to an event storage area in the data storage device. During an execution phase of the data storage device initialization, the events are retrieved and executed by the microprocessor to cause the files having at least the first flag set to be written in a compacted manner to storage locations in a second storage area of the memory device to maintain the integrity of those files. Subsequently, the first storage area of the memory device is erased to provide additional storage space for the writing of new files to the memory device. If a power failure or system failure occurs during the two-phase initialization process, the process is re-started generally at the point where it was interrupted so as not to leave the data storage device in a partially compacted, and hence, a potentially inconsistent state.

Current US Cross Reference Classification (1):
713/2

First Hit Fwd Refs

End of Result Set

☐ Generate Collection Print

L26: Entry 3 of 3 File: USPT Dec 7, 1993

DOCUMENT-IDENTIFIER: US 5269022 A
TITLE: Method and apparatus for booting a computer system by restoring the main memory from a backup memory

Abstract Text (1):
In a computer system, when the system is first booted in a normal mode, main memory data stored in a main memory immediately after the system is booted, is stored as backup data in a backup memory or the like. A backup flag representing whether or not the backup data can be restored is set and the system is rebooted. When the system is next booted in the normal mode, the backup data stored in the backup memory or the like is restored as the main memory data in the main memory. The backup flag is automatically reset in a maintenance mode.

Current US Original Classification (1):
713/2

First Hit Fwd Refs

End of Result Set

Generate Collection Print

L5: Entry 1 of 1 File: USPT Aug 13, 2002

DOCUMENT-IDENTIFIER: US 6434696 B1
TITLE: Method for quickly booting a computer system

Brief Summary Text (7):
There two kinds of boots; "cold boots" and "warm boots", which rely on the state of the computer system when the boot operation is requested. A "cold boot" is performed when power is applied to the computer or a reset button is pressed. When an operating system is loaded in memory already and the computer system is powered on already, a user may request a "warm boot" by entering a predefined sequence of key strokes, e.g., <Ctrl>+<Alt>+<Del>. The BIOS codes include a plurality of computer routines for controlling devices such as a system clock, video output display 6, disk controller 5, and keyboard and thus provide a low-level interface to these devices. The BIOS is generally stored in a Flash ROM.

Brief Summary Text (9):
When a "warm boot" is requested or a reset button is pressed, it is desirable that the time required for the boot process is reduced to force the computer into a ready state as quickly as possible. The boot process is usually called "quick boot", which is achieved by simplifying some device diagnosis processes or loading the device status information that was obtained at the preceding boot time from a storage medium such as disk. Because the quick boot means a boot process in which some POST operations, e.g., memory test are skipped, the quick boot is generally referred to as "quick post".

Detailed Description Text (5):
The method for saving the boot configuration information to a disk will be described now in detail referring to FIG. 4. When power is turned on or a reset button is pressed (S31), a cold boot or warm boot is requested. The POST operation is, first, executed (S32) and then an INT 19h service routine is called to load an operating system (S33). By calling the INT 19h, control is passed to a bootstrap loader which loads the operating system into a memory to prepare the personal computer for use.

Detailed Description Text (11):
Once power is turned on or reset button is pressed (S51), a quick POST operation including skip of memory test is executed (S52), and then it is checked whether or not there is any boot configuration information that has been saved to a disk in the preceding boot process (S52-1). If it is determined that a boot configuration information exists, the operation for its restoration is performed (S53).

L1: Entry 1 of 9 File: USPT Feb 5, 2002

DOCUMENT-IDENTIFIER: US 6345294 B1
TITLE: Methods and apparatus for remote configuration of an appliance on a network

Detailed Description Text (279):
For state that requires update (the database, the cache, etc.) multiple small partitions are used. This organization reduces the chance of the entire disk becoming inconsistent. In the worst case, only one or two partitions are inconsistent. This organization has the additional benefit that in most cases reboot is fast, because only one or two partitions need to be checked.

L1: Entry 4 of 9 File: USPT Nov 2, 1999

DOCUMENT-IDENTIFIER: US 5978913 A
TITLE: Computer with periodic full power-on self test

Brief Summary Text (8):
The present application discloses a computer system which usually performs a reduced set of Power-On-Self-Test (POST) operations during the boot process, but which also spontaneously launches a more extensive set of POST operations during some reboots. This provides fast reboots most of the time, while also providing a full diagnostic often enough to detect system problems early. Preferably the full set of POST operations is launched whenever a certain number of days has elapsed since the last full set of POST operations, and the number of days between POST operations is itself a user-programmable value.

First Hit

Generate Collection | Print

L17: Entry 2 of 3 File: DWPI Nov 28, 2000

DERWENT-ACC-NO: 2001-209975
DERWENT-WEEK: 200121

TITLE: Program storage device with computer readable code for sharing flash ROM in computer system, stores codes for granting processor access to resource after microcontroller has been idled and for releasing resource from processor

Basic Abstract Text (7):
ADVANTAGE - Facilitates setting of remap bit to indicate that the ROM BIOS content has been copied into main memory array, after shadow operation, so that it enables the remapper to deflect accesses to flash ROM, typically residing at FFFFXXXXh, to the shadowed memory locations preferably located at 000FXXXXh. Thus, warm reboots can be executed from the shadowed ROM BIOS which eliminates conflicts with the microcontroller. By sharing the resource flash ROM, system cost is reduced, the system reliability is enhanced and accessibility to flash ROM after boot-up process is improved, reliably.

L14: Entry 1 of 4 File: USPT Apr 6, 2004

DOCUMENT-IDENTIFIER: US 6718462 B1
TITLE: Sending a CD boot block to a client computer to gather client information and send it to a server in order to create an instance for client computer

Abstract Text (1):
Client Discovery is a program that utilizes the existing remote boot capability of a network and adds a program to automate the discovery of a computer's class and the creation of an instance for that specific computer. When a client machine is connected to the network and turned on for the first time, the server will find its request for an operating system, identify that the machine does not have an instance and will send an CD boot block to the client computer. The CD boot block will obtain information about the client computer by automatically scanning everything that the program is able to scan. The client information gathering program may also prompt the user with questions and include those responses in the client information. The CD boot block will then send the information gathered to a log file in the server. One or more keys from the client information are compared to one or more keys in the template table. When a match is made, the server program executes a command to link the client machine's MAC address to the template creating an instance for the client computer and to save the instance in the server memory. The client machine will then reboot. When it restarts it will now be automatically identified by its instance and the appropriate operating system sent to it.

L14: Entry 2 of 4 File: USPT Feb 10, 2004

DOCUMENT-IDENTIFIER: US 6691165 B1
TITLE: Distributed server cluster for controlling network traffic

Abstract Text (1):
A scalable, distributed, highly available, load balancing server system having multiple machines is provided that functions as a front server layer between a network (such as the Internet) and a back-end server layer having multiple machines functioning as Web file servers, FTP servers, or other application servers. The front layer machines comprise a server cluster that performs fail-over and dynamic load balancing for both server layers. The operation of the servers on both layers is monitored, and when a server failure at either layer is detected, the system automatically shifts network traffic from the failed machine to one or more operational machines, reconfiguring front-layer servers as needed without interrupting operation of the server system. The server system automatically accommodates additional machines in the server cluster, without service interruption. The system operates with a dynamic reconfiguration protocol that permits reassignment of network addresses to the front layer machines. The front layer machines perform their operations without breaking network communications between clients and servers, and without rebooting of computers.

L14: Entry 3 of 4 File: USPT Dec 7, 1999

DOCUMENT-IDENTIFIER: US 6000030 A
TITLE: Software fingerprinting and branding

Abstract Text (1):
A method and apparatus for controlling the distribution of computer software products stored at a file server provide for requesting the identity of the user and the user's secret key prior to enabling access to a requested program product. The program product, upon proper verification of the user identify, is encoded using a second key which is known to the user, and preferably an identification of the user is embedded in the encoding program. Various methods are employed for tracking user access to particular programs, including storing the identify of the user either camouflaged in a commonly found program in non-volatile memory or hidden in a typically overlooked portion of non-volatile memory. In addition, the encoded program can have embedded therein one, and preferably two identifications of the user which can be used to track the program as well as to ensure that the program, when executed has not been tampered with. Upon execution of the encoded program by the user's system, the code is executed "on the fly" and no executable copy of the code is stored in non-volatile memory at any time. In the event that decoding does not result in a properly executable program, the system may need to be rebooted and/or the program may be at least partially destroyed.

First Hit

End of Result Set

Generate Collection | Print

L14: Entry 4 of 4 File: DWPI Dec 4, 2003

DERWENT-ACC-NO: 2004-080940
DERWENT-WEEK: 200408

TITLE: Reboot tracking system for distributed system, has central server collecting log files from file server and storing records in database to generate reboot status report based on records in database

Basic Abstract Text (1):
NOVELTY - The system (100) has a start-up routine (150) configured to create a record reflecting a reboot status associated with one of multiple terminals (110, 112, 114). A file server receives and stores the record in a log file from the terminal. A central server (130) collects the log files from the file server and stores the records in a database to generate a reboot status report based on the records.

L13: Entry 1 of 4 File: USPT Dec 17, 2002

DOCUMENT-IDENTIFIER: US 6496942 B1
TITLE: Coordinating persistent status information with multiple file servers

Detailed Description Text (29):
In a REBOOTING state 240, this file server 110 has control of none of the mass storage devices 120 and is recovering from a service interruption.

Detailed Description Text (58):
In this state, this file server 10 performs no operations, until this file server 110 determines that it reboot.

Detailed Description Text (62):
In the REBOOTING state 240, this file server 110 has control of none of the mass storage devices 120 and is recovering from a service interruption.

Detailed Description Text (64):
If this file server 110 is unable to recover from the service interruption, the REBOOT-FAILED transition 241 is taken and this file server 110 remains in the REBOOTING state 240.

Detailed Description Text (65):
If this file server 110 is able to recover from the service interruption, but the other file server 110 is in the TAKEOVER state 220, the REBOOT-FAILED transition 241 is taken and this file server 110 remains in the REBOOTING state 240. In this case, the other file server 110 controls the mass storage devices 120 normally assigned to this file server 110, and this file server 110 waits for the GIVEBACK-OPERATION transition 221 before re-attempting to recover from the service interruption.

Detailed Description Text (78):
At a step 258, if this file server 110 was in the NORMAL state 210 before entering the REBOOTING state 240 (that is, this file server 110 performed the step 254 and seized only its own mass storage devices 120), it enters the NORMAL state 210.

Detailed Description Text (79):
At a step 258, if this file server 110 was in the TAKEOVER state 220 before entering the REBOOTING state 240 (that is, this file server 110 performed the step 255 and seized all the mass storage devices 120, it enters the TAKEOVER state 220.

L13: Entry 3 of 4 File: USPT May 20, 1997

DOCUMENT-IDENTIFIER: US 5631847 A
TITLE: System for network file server failure notification

Detailed Description Text (29):
An example of a correction technique is reboot. It is well known in the computer arts that many errors can be corrected by simply triggering a reboot signal, which causes the rebooted device (file server 1) to "start over" and reload the BIOS and operating system, and in some instances, other operating instructions. Reboot, which can, in some instances, be done by momentarily interrupting power at some point in the system, is then a viable correction technique for many error conditions, and is one technique programmed in routines 18, which a responsible party could cause to be activated by remote input.

L13: Entry 4 of 4 File: USPT Jan 7, 1997

DOCUMENT-IDENTIFIER: US 5592611 A
TITLE: Stand-in computer server

Brief Summary Text (9):
It is also known to provide a replacement server, for instance when a server has suffered a failure that cannot be repaired. In such replacement, the software environment of the failed server is recreated, by copying the software of the failed server (and, for instance, to restore a backup of the files from the failed file server) to the replacement server, and to reboot the replacement server under the name of the failed server.

First Hit Fwd Refs

End of Result Set

☐ Generate Collection Print

L3: Entry 1 of 1 File: USPT Mar 23, 2004

DOCUMENT-IDENTIFIER: US 6711675 B1
TITLE: Protected boot flow

Detailed Description Text (31):
1) Determine reboot--The processor can be reset for multiple reasons. These reasons include waking from a power-saving sleep state, partial reboot, warm reboot, cold boot, and others. The boot sequence may be somewhat altered depending on which type of reboot is being executed.

Detailed Description Text (59):
8) Perform cold boot--Initiate a full reboot.

L28: Entry 1 of 7 File: USPT May 4, 2004

DOCUMENT-IDENTIFIER: US 6732264 B1
TITLE: Multi-tasking boot firmware

Detailed Description Text (5):
To facilitate booting the system the first time after it is manufactured, a flag in nonvolatile memory can optionally be used to signal the BIOS boot code that the system hardware has not been enumerated yet. For example, when the BIOS program code is programmed into flash memory on the assembly line, a "first boot" flag can be set in the flash memory. When the BIOS boot code checks for hardware changes at 14 in FIG. 1, it checks this flag. If the first boot flag is clear, it proceeds to check the hardware latch. If the first boot flag is set, it clears the flag and proceeds to enumerate the hardware at 16. This check for a first boot flag can be performed in addition to the check to see if the case has been opened, or as an alternative. That is, 14 in FIG. 1 can include either of these checks, or both of them, depending on which features are available on the hardware.

L28: Entry 2 of 7 File: USPT Apr 13, 2004

DOCUMENT-IDENTIFIER: US 6721881 B1
TITLE: System and method for determining if a display device configuration has changed by comparing a current indicator with a previously saved indicator

CLAIMS:

5. The method of claim 1 further comprises: determining whether a simple boot flag register is available; and performing, if the simple boot flag register is not available, a full video display detection routine.

10. The system of claim 6 further comprises: a module that determines whether a simple boot flag register is available; and a module that performs, if the simple boot flag register is not available, a full video display detection routine.

16. The computer-readable medium as recited in claim 11, wherein the computer program further includes instructions for: determining whether a simple boot flag register is available; and performing, if the simple boot flag register is not available, a full video display detection routine.

L28: Entry 3 of 7 File: USPT Jan 27, 2004

DOCUMENT-IDENTIFIER: US 6683528 B2
TITLE: Portable computer supporting paging instructions

CLAIMS:

10. A method of securing a computing device, the method comprising: executing a boot program in the device, the boot program comprising instructions which when performed by a microprocessor performs a boot sequence comprising the steps of: retrieving a resettable status flag from a memory associated with a wireless receiver, the flag having at least first and second states; completing boot sequence performance if the flag is in the first state; and terminating boot sequence performance if the flag is in the second state.

L28: Entry 3 of 7 File: USPT Jan 27, 2004

DOCUMENT-IDENTIFIER: US 6683528 B2
TITLE: Portable computer supporting paging instructions

CLAIMS:

10. A method of securing a computing device, the method comprising: executing a boot program in the device, the boot program comprising instructions which when performed by a microprocessor performs a boot sequence comprising the steps of: retrieving a resettable status flag from a memory associated with a wireless receiver, the flag having at least first and second states; completing boot sequence performance if the flag is in the first state; and terminating boot sequence performance if the flag is in the second state.